Automated Software Engineering

What We Do

We are the Automated Software Engineering group, based at NASA's Ames Research Center on Moffett Field, California. The goal of automated software engineering is to increase by orders of magnitude both the quality and the productivity of software engineering. The cross-cutting research done by the Automated Software Engineering group of Code IC at NASA Ames draws upon several disciplines including: Artificial Intelligence - particularly automated reasoning and knowledge representation; Formal Methods; programming language theory; mathematical logic; and advanced compiler methods. Our focus is on strategic research; i.e., research that is directed to the 5 - 15 year time horizon, aiming to make large impacts rather than incremental advances. Our research is done in the context of pacing NASA applications, as a means of both providing us feedback and as a means for us to make contributions to NASA's goals as we progress. We currently have space-related projects in space science code generation, and in software verification for deep-space missions. For civilian aviation, we are engaged in research on next-generation autocoding technology and high-assurance software design. We are also developing innovative educational technology, and we have some sample lessons for students and teachers based on work done so far.

Current Highlights

"Automating the Documentation and Certification of NASA Software" May 2004.
Ewen Denney
The ASE group is developing automated program synthesis systems for the application domains of data analysis (AutoBayes) and state estimation (AutoFilter). We have previously extended these systems with an automated certification capability, based on mathematical logic, for various safety policies. However, it is very difficult for humans to interpret the resulting proofs and then relate them to the original program. We have addressed this by incorporating a safety documentation feature: a tool that can automatically generate textual explanations of safety with respect to a given safety policy for auto-generated code. This technology has the potential to increase confidence in the use of code generators within and outside NASA. Auto-generated code, in addition to a certificate of correctness (w.r.t. user-defined notions of safety) will come with a document containing human-readable explanation as to why it is correct. This approach is a signific
Presentations [(**PDF, 176K**),(**PostScript, 640K**),(**PowerPoint, 253K**)]

"Simulation-Based Verification of Autonomous Controllers via Livingstone Path Finder." May 2004.
Charles Pecheur and Tony Lindsey
Model-based autonomous systems pose hard verification challenges because of the huge space of unpredictable circumstances in which they are expected to operate. Livingstone 2 (L2) is a model-based diagnosis system developed at NASA Ames, intended for autonomy applications. We developed Livingstone Pathfinder (LPF), an advanced verification tool for Livingstone 2. LPF features a scripting language for specifying, in a flexible and concise way, a large set of execution scenarios to be analyzed. The search engine explores the tree of possible executions according to different strategies, searching for discrepancies between actual and diagnosed state. We have successfully applied that technology on a real-size model (X-34 propulsion feed subsystem, 800+ variables, bottom two pictures). This case study demonstrated the scalability of the approach to large applications, and also detected an error that lead to a minor adjustment of the X-34 model.
Presentations [(**PDF, 821K**),(**Postscript, 1.4M**),(**Powerpoint, 431K**)]

CLICK HERE TO VIEW MORE HIGHLIGHTS

Last changed: 17 December 2004 by Raymond De Ocampo